Multi Method Local Path Planning Considering Flight Dynamics Constraints for Collision Avoidance

L. Danielmeier*, S. Seitz, M. Hartmann, N. Voget and D. Moormann
RWTH Aachen University, Wüllnerstraße 7, Aachen, Deutschland

ABSTRACT

The regulation for automatic UAV flights in Europe requires a UAV to be able to avoid collisions. Ideally, the avoidance path of the UAV should be explainable for pilots flying the aircraft to be avoided. However, accomplishing this using only one avoidance method can be difficult. Methods that generate a path that is understandable by humans are often not capable of solving complex avoidance scenarios. On the other hand, using a complex avoidance method for every avoidance scenario can result in unexplainable avoidance paths for simple avoidance scenarios. Therefore, this paper presents a local path planning concept for collision avoidance, using four different path planning methods in combination with an avoidance method selection logic. These are vertical avoidance, avoidance by deviating to a path that is parallel to the original flight path, varying the speed on the original path, and the mCOWEX algorithm. This yields simple and understandable avoidance paths whenever possible, while at the same time being able to fall back to the high success rate of the more complex methods.

1 INTRODUCTION

The demand for UAV capable of fully automatic BVLOS operation is steadily increasing due to their versatility in many different use cases such as transport of goods or search and rescue missions. For legal BVLOS operation of automatic UAVs in the EU a fully automatic, robust, but also deterministic collision avoidance is required. In this context, deterministic means, that given a similar avoidance situation, i.e. two slightly different head-on collisions, the UAV should still use a similar avoidance path for both evasions, as a human operator would. This is required, so that a human pilot encountering the UAV as a collision partner, is able to predict what path the UAV will take based on the collision scenario. For some approaches to automatic collision avoidance, such as A* [1], COWEX [2], mCOWEX [3] or potential field methods [4] this poses a problem, as the exact same avoidance scenario will generate the exact same avoidance path, but a slightly different scenario might yield a vastly different path. The difference might be as small as a meter offset in the intruder position relative to the UAV. This is problematic for the explainability of the avoidance maneuver, as a human would be unable to predict the avoidance path the UAV might take for a given scenario. On the other hand, the avoidance method cannot be so simple, that it will not be able to find a collision free path in a more complex avoidance scenario.

For an avoidance path to be successful, it must not only be free of collisions, but it must also be flyable with respect to the flight dynamics constraints, such as minimal turn radius, acceleration and deceleration capability as well as maximum climb and decent rates. There are two approaches to consider these flight dynamics constraints of the aircraft for avoidance path planning. In the first approach, algorithms that do not consider the constraints are used to plan a number of avoidance paths. These are checked for flight dynamics compliance after they are generated and the shortest path that complies to the flight dynamics constraints is used while those who do not comply are discarded. This is not computationally efficient, as the available computing time is used for paths that are not flyable and have to be discarded. In other approaches, methods and algorithms that consider the flight dynamics constraints during planning such as constrained A* [5], constrained RRT [6] or mCOWEX are used. These algorithms have a higher computation time in comparison to unconstrained algorithms, but only have to compute one successful path. Thus the available computation time is used more efficiently.

Therefore, this paper proposes a multi method local path planner to generate explainable avoidance paths for simple situations using simple avoidance methods, while being capable of solving complex avoidance scenarios using the more complex mCOWEX algorithm.
First, the four collision avoidance methods used in this paper are explained. Second, the logic, with which the path generation method is chosen based on the avoidance scenario is presented. Third, Monte Carlo simulations for different sets of avoidance scenarios are conducted with each method separately as well as the combination of all methods i.e. the multi method local path planner. The results are then used to compare the success rates of each standalone method with the success rate of the multi method local path planner.

*Email address: danielmeier@fsd.rwth-aachen.de

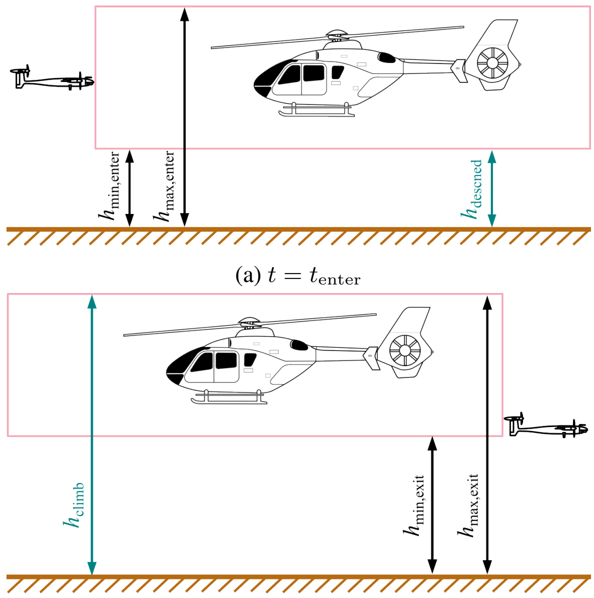
Thereafter, examples of avoidance paths generated using each of the methods of the multi method local path planner will be shown. The results presented in this section will also be briefly discussed.

2 COLLISION AVOIDANCE METHODS

This section will give an overview of the four collision avoidance methods used in the multi method local path planer presented in this paper. As these methods are path planning algorithms they all rely on an estimation of the intruder position at a given time. This position is estimated using a 99% confidence interval of the possible positions of the intruder given by the covariance matrices of an IMM Kalman Filter, i.e. [7]. This area of possible intruder positions is displayed by the dark red ellipse i.e. Fig. 7. The light red ellipse around the area of possible intruder positions is a safety zone of a predefined size. This safety zone extends to 300m horizontally and +/-50m vertically. The safety zone for static obstacles is displayed by a light red circle around the dark red obstacle.
The UAV's position, speed and heading on a given path are calculated by the UAV's flight guidance system, therefore these are also assumed to be known [3]. During flight a collision detection system continuously monitors the information about intruders, supplied by a different systems such as FLARM and ADS-B. A collision is defined as the loss of a defined minimum separation between any obstacle or any intruder and the UAV. If a possible collision has been detected, the local path planer is triggered and all four path planning algorithms are executed in parallel. Thereafter, all valid paths are stored and the the avoidance method selection logic is used to select a path. This selection process will be explained in Sec. 3.

2.1 Vertical Avoidance

The first method of collision avoidance used in this paper alters the height profile of the original path to avoid collisions. This offers the advantage that the geofence and other static 2D obstacles, that were considered in the original path, do not have to be considered while generating the avoidance path. A disadvantage of altering the height profile, is that information about the ground height of the surrounding terrain has to be considered and processed for the avoidance path generation. To generate a collision free height profile, the times t_{enter} and t_{exit} and the positions p_{enter} and p_{exit} at which the UAV enters and leaves the safety zone of the intruder are calculated. Then, the maximum and minimum heights of the intruder's vertical safety zone for the duration between entry and exit are determined. Using these parameters, the altitudes h_{climb} and h_{descent} required for a collision free height profile are derived, see. Fig. 1. To guarantee, that no airspace infringements or ground collisions occur, these altitudes are then checked for compliance with the ground profile, as well as all airspaces crossed by the flightpath. If there is a conflict with the ground or the airspace, the maneuver can not be executed.



(a) $t = t_{\text{enter}}$

(b) $t = t_{\text{exit}}$

Figure 1: Visualization of the altitudes h_{climb} and h_{descent} at $t = t_{\text{enter}}$ and $t = t_{\text{exit}}$.

The following steps are the same for the climb and the descent maneuver, therefore they are only explained for the climb maneuver.
First, the point on the original path, where the climb must begin in order to reach h_{climb} two seconds before t_{enter} is calculated using the maximum climb rate of the UAV. If the UAV has already passed this point the maneuver is not flyable and therefore either the descent maneuver or a different approach must be used. After the climb, altitude is held at h_{climb} from $t_{\text{enter}} - 2\,s$ to $t_{\text{exit}} + 2\,s$ and thereafter the UAV descends with maximum descent rate until the height profile of the original path is reached.

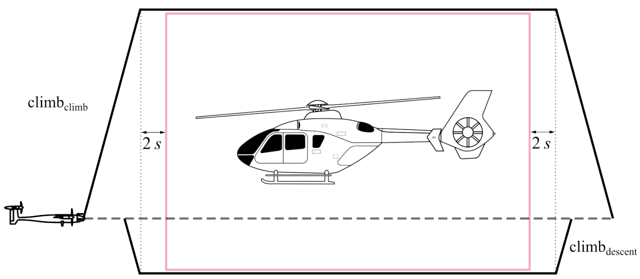


Figure 2: Visualization of climbs required for the climb and the descent maneuver.

If both the climb and descent maneuver are possible, the required climb timesfor both maneuvers are compared. As climbing is more energy consuming than descending, and both maneuvers can have a climb and a descent, the maneuver with the smaller climb is chosen. This is shown in Fig. 2, where the required climb for the climb maneuver is much larger than the required climb for the descent maneuver.

2.2 *Avoidance using Parallel Paths*

Within the second method a flight path parallel to the original flight path is used to generate a collision free avoidance path. The first step for generating a parallel avoidance path is to determine to which side the path should deviate from the original path. This is done by analyzing the avoidance scenario and then choosing a side to avoid the collision according to SERA.3210, the **S**tandardized **E**uropean **R**ules of the **A**ir [8]. These rules define how a potential conflict between two aircraft should be resolved for a given conflict scenario, as shown in Fig. 3.

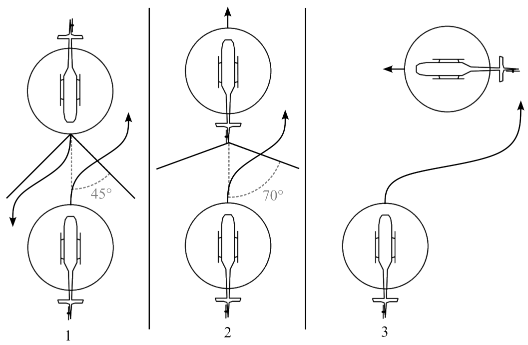


Figure 3: Right-of-Way according to the SERA.3210.

In the scenarios shown above the aircraft with the right-of-way can continue on its path while the other aircraft has to take steps to avoid it. As the aircraft used in this paper is an unmanned and fully automatic UAV it never has the right-of-way and must always take steps to avoid the collision [9]. In cases 1, 2a, 2b and 3a this does not contradict SERA.3210 and gives even more safety margin between the intruder and the UAV as seen in Fig. 4. The orange arrow in 2b shows the path of the UAV according to SERA.3210 and the green arrow shows the path the UAV must take in this scenario.

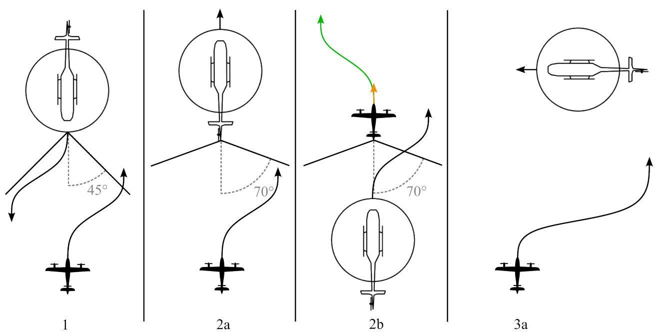


Figure 4: SERA.3210 for UAV, cases 1, 2a, 2b and 3a.

In case 3b and case 3c however the outcome of the conflict scenario depends on the action taken by the intruder. If the pilot of the intruder aircraft recognizes that the UAV is an UAV and therefore it must take evasive action even though it has right-of-way according to SERA.3210, he will remain on course and not take evasive action himself. In this case the best evasive action for the UAV would be to move to a parallel path to the left, see case 3c in Fig. 5. If however the pilot does not feel safe in this assumption, or does not know that the UAV must always avoid, he might take evasive action according to the SERA.3210 himself. In this case the best avoidance path would be a path to the right, see case 3b in Fig. 5.

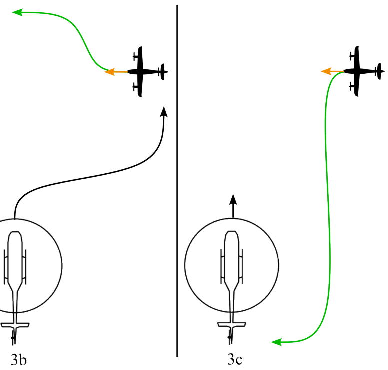


Figure 5: Dilemma in cases 3b and 3c.

As this case has no clear correct solution, other avoidance methods must be used. Otherwise, once the appropriate side for the avoidance maneuver has been determined, a parallel path is planned.

The path is divided into three subpaths: two connecting path segments (1&3) and a parallel path segment (2) as seen in Fig. 6. All parts of the path are planned with the speed v_{UAV}, the UAV is flying at the moment the avoidance path planning is triggered.

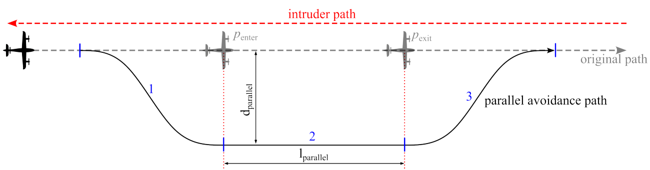


Figure 6: Subpaths of a parallel avoidance path.

The connecting path segments (1&3) are generated using Dubins paths from the current position to the start of the parallel path segment and from the end of the parallel path segment to the rejoin point on the original path [10]. These Dubins paths are planned using the minimum turn radius of the UAV at the given speed. To generate the parallel path segment, the time t_{enter}, position p_{enter} and the heading ψ_{enter} at which the UAV would first enter the intruder's safety zone if it would proceed on the original path as planned are determined. Next, the time t_{exit} and the position p_{exit}, at which the UAV would leave the intruder's safety zone is calculated. Finally the size of the safety zone of the intruder at t_{exit}, $d_{\text{safety,max}}$ is calculated. Using these parameters and the avoidance side, the parallel path segment is generated by placing a straight path segment of the length $l_{\text{parallel}} = v_{\text{UAV}} \cdot (t_{\text{exit}} - t_{\text{enter}})$ with the heading ψ_{enter}. This straight path segment is then positioned such that its starting point is at the distance $d_{\text{parallel}} = d_{\text{safety,max}}$ to the left or the right of p_{enter}.

The parallel path segment is then combined with the connecting path segment. The the second connecting path intersects the original path on a point located one second behind p_{exit}. As p_{exit} is the point from where the original path is collision free, this guarantees a collision free path after rejoining.

The finished parallel avoidance path is checked for validity i.e. is checked for collisions with the intruder as well as other static obstacles and geofence breaches. If the parallel avoidance path is not collision free, due to static obstacles or a second intruder, the distance d_{parallel} is increased by 10% of $d_{\text{safety,max}}$ up to ten times. As this method is not designed for more than one intruder or static obstacles, the generation of the path does not consider them. However, by incrementally increasing the distance of the parallel path segment away from the original path collision free paths are achievable with this method even for such scenarios. Once a generated path is collision free it is chosen as the final avoidance path.

2.3 Avoidance using Speed Variations

The third method used, is the variation of the speed at which the path is flown. This method therefore does not deviate from the original path at all. This poses the advantage, that neither the geofence nor airspaces or ground height have to be taken into consideration for the avoidance maneuver, as these where already taken into account while planning the original path. The disadvantage however is that this method can only solve a very specific set of collision scenarios. Any collision where an intruder's heading is pointing at the UAV i.e. a head-on or a overtaking scenario cannot be solved by varying the speed of the UAV. However, scenarios such as the intersection of the UAVs path by the intruder can be solved very effectively by using this method.

To check whether a speed change can solve the collision, the possible reachable speeds of the UAV starting at the point where the collision avoidance is triggered are calculated in 1m/s increments. This contains decelerations and accelerations. For each of these speeds, starting at the speed closest to the current UAV speed, the movement of the UAV along the path with the new speed as well as the corresponding movement of the intruders is simulated. This is done until either the simulated UAV enters the safety zone of one of the simulated intruders or all simulated intruders start to move away from the simulated UAV. If the simulated UAV enters the safety zone of a simulated intruder, the tested speed is not suitable to avoid the collision and the next speed must be tested. If all simulated intruders start to move away from the simulated UAV and no collision has occurred until that point, the speed checked is suitable to avoid the collision. The point at which all simulated intruders start to move away from the simulated UAV is then used as the point at which the UAV can start to accelerate or decelerate to the speed originally planned on that part of the path.

2.4 Avoidance using mCOWEX Algorithm

The final collision avoidance method used for multi method local path planner is the mCOWEX algorithm. The detailed explanation of the algorithm can be found in [3]. This algorithm is capable of avoiding collisions with more than one intruder as well as multiple static obstacles. The downside is, that the avoidance path is determined only by the cost function and can therefore differ from avoidance paths that a pilot might take, as seen in Fig. 7. Therefore, the algorithm is only used in scenarios with more than one intruder or if there is an intruder in combination with static obstacles.

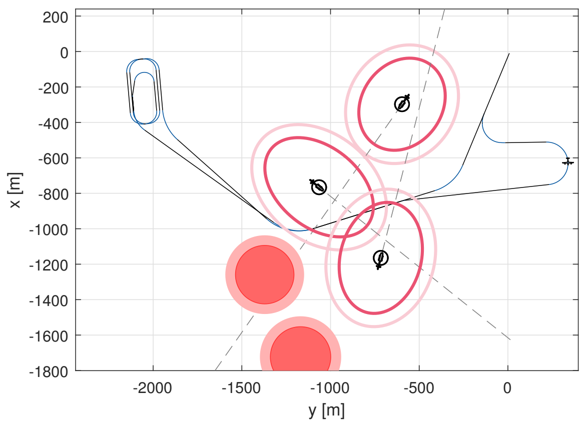
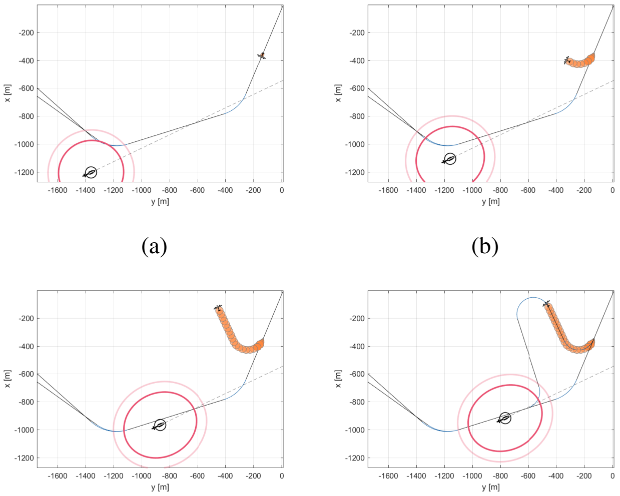


Figure 7: A mCOWEX avoidance path for three intruders and two obstacles, that does comply with SERA.3210, but is collision free.

The mCOWEX algorithm works by expanding circles (denoted as waves from here) into collision free 2D-space within the geofence. This is done by first expanding a wave around the starting point and thereafter expanding new waves on the boundary of an existing wave. Which wave is expanded next is determined by a cost function. The placement of the newly expanding waves, is constrained by the minimal turn radius of the UAV, as the centers of the wave determine the waypoints of the final path. The algorithm stops once the endpoint can be connected to the current wave with a collision free Dubins path. An example of the generation of a collision free path with the mCOWEX algorithm for one intruder is shown in Fig. 8.

If the mCOWEX algorithm does not find a collision free path at the current UAV speed, it is checked to what speed the UAV can decelerate on the original path and still have enough separation from the intruder to complete a 90° turn. A slower speed gives the UAV a smaller minimum turn radius and therefore increases the mCOWEX algorithms chances of success. The mCOWEX algorithm is then restarted with the new starting parameters.



(a) (b) (c) (d)

Figure 8: Generation of a path with the mCOWEX algorithm for a head-on collision.

3 AVOIDANCE METHOD SELECTION LOGIC

The logic used to select which of the four methods presented above is used if more than one method provides a collision free and valid path will be explained in this section. Once the local path planer is triggered all methods will simultaneously calculate a path. Once all methods have finished, their path is added to the pool of possible paths, if they have found a valid collision free path. If there is more than one valid path, a criterion has to be used to select which path is best suited for avoiding the collision. The most important criterion when considering possible avoidance paths is to create a successful separation between the UAV and all intruders, obstacles, airspaces, geofences and the ground. As this is already done by the four presented methods a secondary criterion must be found to distinguish between the available paths.

For the multi method local path planner the second criterion is to choose the path that most resembles a human pilots reaction. This criterion is chosen, to simplify the prediction of the future UAV path for human pilots of intruders as well as ATC controllers. If the UAV for example always uses the mCOWEX algorithm first, which finds a path in almost all scenarios, it will theoretically always fly on a safe path. Since this path is defined by a cost function the path may seem completely erratic to the pilot of the intruder aircraft causing him to fly the intruder plane into a more dangerous situation.

Using this second criterion the paths are selected in the following order: Vertical Avoidance, Parallel Paths, Speed Variation and mCOWEX Algorithm. First, if possible, the path using the vertical avoidance method is chosen. As a pilot of an intruder is easily able to distinguish that the UAV is no longer flying at the same altitude as himself, this method requires the least amount of guesswork by the pilot, if the UAV is indeed on a collision free path. Second, the parallel path method is chosen. This method follows the SERA, so a pilot should also be able to predict the flight path taken by the UAV and act accordingly. Next, the speed variation method is chosen. This does not resemble a conventional avoidance path. It does however follow the original path of the UAV and does not include sudden changes of direction, which may be hard to understand for pilots. If all these paths are not valid, the path generated by the mCOWEX algorithm is used. This path does not consistently comply with SERA, but always provides a safe path, given that all intruders continue flying as they have done at the point the collision avoidance is triggered.

If none of the paths is valid, the same procedures are deployed again using only half of the original safety zone around the intruders. This causes a closer separation but will still guarantee a collision free flight path. If reducing the safety margin does not generate a valid path, the UAV will trigger its flight termination system, which stops all engines and deploys a parachute, to guarantee that no collision occurs.

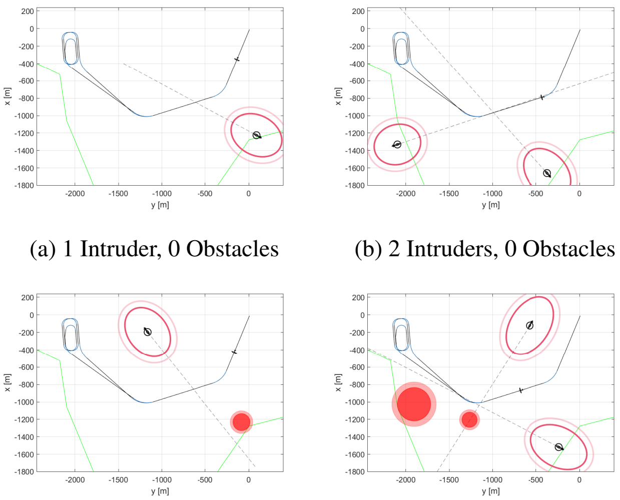
4 EVALUATION AND DISCUSSION

To evaluate the presented multi method local path planner, first all four path planning methods separately as well as the combination of all four methods will be evaluated by running Monte Carlo simulations with different numbers of intruders and obstacles. The results of these simulations will be used to evaluate the success rates of the different methods for a given avoidance scenario. Thereafter, examples of paths planned for different avoidance scenarios using the multi method path planner will be shown and briefly discussed.

4.1 Success Probability of the Different Path Planning Methods

The Monte Carlo simulations were conducted with four different combinations of zero to two obstacles and one to two intruders. Examples of the scenarios used can be seen in Fig. 9. For the simulation, the starting positions and safety zones[1] of all obstacles and intruders, as well as the intruder's speed and heading were varied. The UAV always flew on the same path, see Fig. 9, with the same starting speed and height profile.

[1]The safety zone defined by a safety radius and is shown as a light red for the obstacles and intruders. The intruder's predicted flight path for the next 60 seconds is shown with a dashed gray line.



(a) 1 Intruder, 0 Obstacles (b) 2 Intruders, 0 Obstacles

(c) 1 Intruder, 1 Obstacle (d) 2 Intruders, 2 Obstacles

Figure 9: Examples of avoidance scenarios with different number of intruders and obstacles. The original path of the UAV is shown in black for the straight segments and blue for the curved segments.

The success probability of the different methods can be seen in Fig. 10. The figures also include the success probability of the multi method local path planner. As the multi method local path planner only needs one of the methods to succeed, its success probability can be higher, than that of the most successful method. For a given scenario, a successful attempt was either successful with the full safety zone (blue) or using a reduced safety zone (orange).
As Fig. 10 shows, the success rate of the first three path planning methods has a strong dependance on the given scenario.
The vertical method scales poorly with the amount of intruders but is not affected by the number of obstacles. The poor success rate with an increasing number of intruders, is caused by a higher chance of encountering an intruder closer to the current position, which in turn decreases the time the UAV has to climb or descend to avoid the intruder. The number of obstacles does not effect the success rate, as this method does not deviate from the original path, and the original path is free of obstacles.
The method of parallel paths has a higher success rate than the vertical method in all scenarios. This is to be expected, as this method deviates from the original path in the horizontal plane and the UAV is more agile in the horizontal plane in comparison to the vertical plane and changes of speed. Therefore, an avoidance path can be found when intruders are to close to climb or descend to avoid them. This method is designed to follow the SERA.3210, which only describes collision avoidance for one intruder, thus it does not offer a good success rate for scenarios with two intruders. As the parallel path method does not consider obstacles while generating the path, this method does not scale well with an increasing number of obstacles.

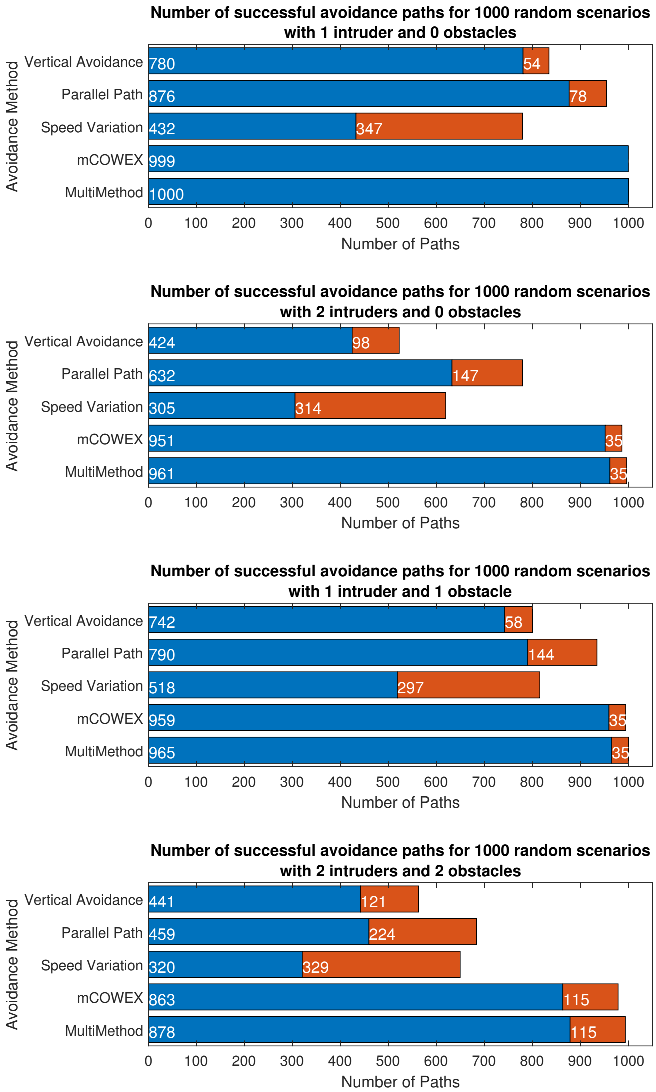


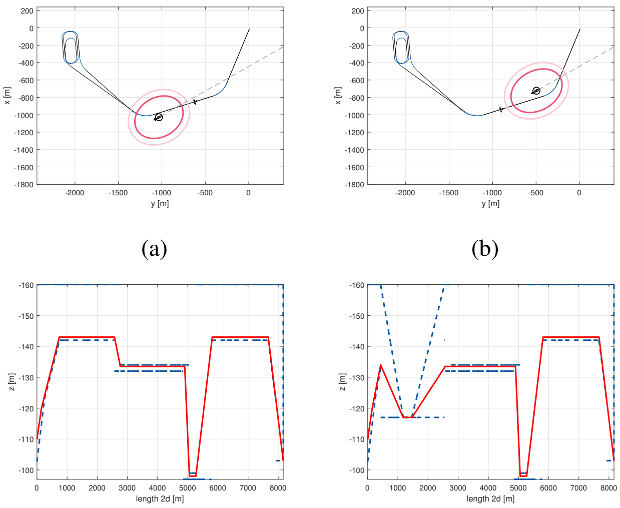
Figure 10: Success rates of all avoidance methods for 1000 randomized avoidance scenarios with different combinations of intruders and obstacles.

The speed variation method scales poorly with the amount of intruders, but is not affected by the amount of obstacles for the same reasons as the vertical method. The greater drop in success rate with the increase from one to two intruders in comparison to the other methods is due to the speed variation method only working for crossing intruders. With the increase from one to two intruders the chances of encountering a head-on or overtaking intruder also double, thus greatly reducing the success rate of the method.

The most successful method is the mCOWEX algorithm, as it takes both intruders and static obstacles into account during path planning.Therefore, its performance only slightly decreases as more obstacles and intruders are added to the scenario. Combining the success rates of all path planning methods to the multi method local path planner, gives a success rate of 99.3% while only needing to reduce the safety margin in 12% of cases. This holds true even in scenarios with two intruders and two obstacles, which are of academic nature, as they are unlikely to occur during real flights. Therefore, the multi method local path planner presents a viable solution to be used as a collision avoidance method for a UAV in automatic BVLOS operation.

4.2 Example Avoidance Scenarios

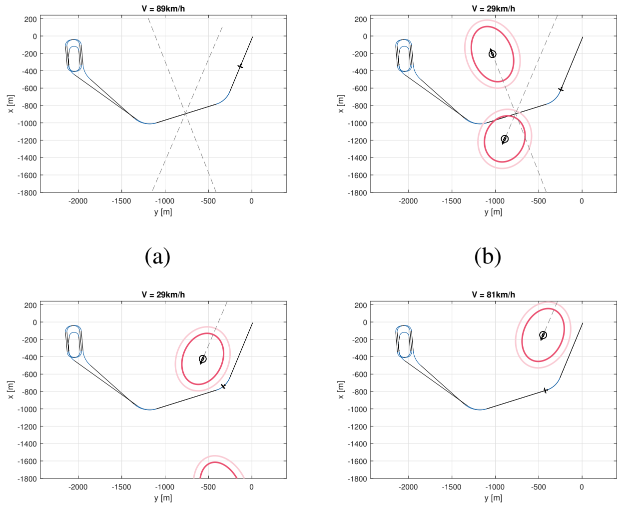
The following section will give an example avoidance path for each type of the presented avoidance scenarios. First, a vertical avoidance path for a scenario with one intruder and no obstacles is shown in Fig. 11.



(a) (b) (c) (d)

Figure 11: Avoidance Path for a scenario with one intruder and no obstacles using vertical avoidance.

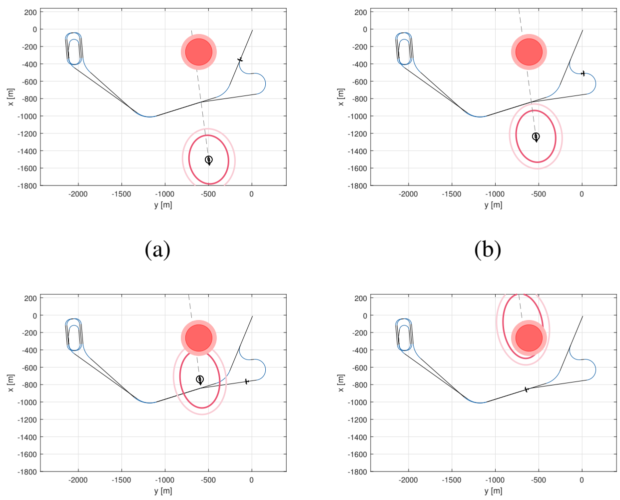
As a vertical avoidance maneuver is difficult to display in 2D, the top view in Fig. 11a and Fig. 11b is expanded by the side view in Fig. 11c and Fig. 11d. The vertical avoidance can be better seen in the side view. The blue lines represent the upper and lower boundaries of the height profile and the red line shows the height at which the UAV flew. In this case the avoidance was accomplished by altering the original height profile, see Fig. 11c, by descending below the intruder between the 2D path lengths of 1200 m and 1500 m, see Fig. 11d. These are the 2D path lengths in which the UAV would have been inside the safety zone of the intruder, if the UAV had followed its original height profile.



(a) (b) (c) (d)

Figure 12: Avoidance Path for a scenario with two intruders and zero obstacles using the speed variation method.

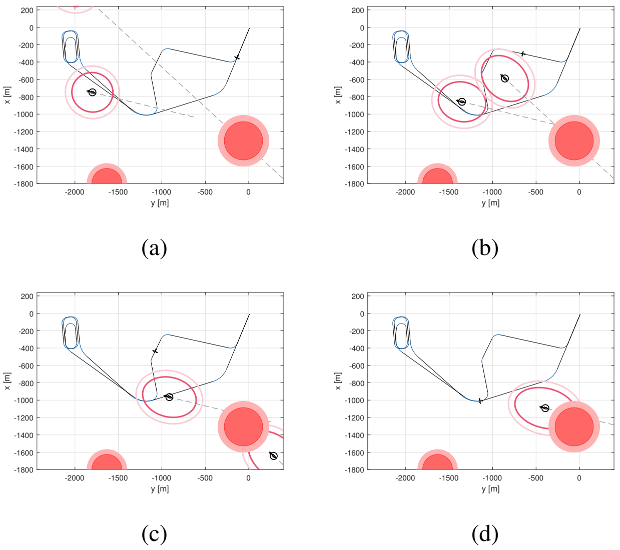
Next, Fig. 12 shows the avoidance of two intruders and no obstacles, using the speed variation method. As seen in the figure, the UAV slows down to avoid both intruders and accelerates to its original speed afterward.



(a) (b) (c) (d)

Figure 13: Avoidance Path for a scenario with one intruder and one obstacle using the mCOWEX algorithm.

Figure 13 shows an avoidance path for a scenario with one intruder and one static obstacle using a path generated with the mCOWEX algorithm. The last avoidance path, see Fig. 14, shows an avoidance path generated by the parallel path method. As the obstacles are on to the left of the UAV and both intruders dictate a parallel avoidance path to the right, the method is able to generate a successful path, even though the scenario is far more complex than what the method was designed to handle.



(a) (b) (c) (d)

Figure 14: Avoidance Path for a scenario with two intruders and two obstacles using the parallel path method.

In conclusion, the multi method local path planner offers a versatile and capable avoidance method, that can deal with complex scenarios, while also generating efficient and understandable paths in simple avoidance situations.

5 CONCLUSION

This paper presents a multi method local path planner that uses a combination of four different collision avoidance algorithms to generate collision free paths. The four methods used are speed variation on the original path, height profile variation on the original path, avoidance to a parallel side path and the mCOWEX algorithm. First, all methods were presented, and their respective pros and cons were given. Second, the logic that determines, which avoidance path is chosen in case more than one method finds a valid path, was explained. Next, the methods were first evaluated individually by evaluating their success rate in different avoidance scenarios, using Monte Carlo simulations. These same simulations were also used to compare the success rate of the combination of all four methods for the same avoidance scenarios. Finally, examples of avoidance paths generated with all methods of the multi method local path planner were given for different avoidance situations. As the multi method local path planner was able to find collision free paths in 100% of all scenarios it is suitable for use in fully automatic UAV in BVLOS operation.

ACKNOWLEDGEMENTS

Special thanks go to the German Federal Ministry for Digital and Transport (BMDV) for funding and to TüV Rheinland Consulting GmbH for support and coordination of the EULE project.

REFERENCES

[1] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[2] I Barz, L Danielmeier, S Seitz, and D Moormann. Kollisionsfreie flugbahnen für kippflügelflugzeuge auf basis des wavefront-algorithmus unter berücksichtigung flugmechanischer beschränkungen. Deutscher Luft-und Raumfahrtkongress, 2020.

[3] Lennart Danielmeier, Sebastian Seitz, Isabelle Barz, Philipp Hartmann, Max Hartmann, and Dieter Moormann. Modified constrained wavefront expansion path planning algorithm for tilt-wing uav. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 676–685. IEEE, 2022.

[4] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings. 1985 IEEE international conference on robotics and automation*, volume 2, pages 500–505. IEEE, 1985.

[5] Zhibo He, Chenguang Liu, Xiumin Chu, Rudy R Negenborn, and Qing Wu. Dynamic anti-collision a-star algorithm for multi-ship encounter situations. *Applied Ocean Research*, 118:102995, 2022.

[6] Beobkyoon Kim, Terry Taewoong Um, Chansu Suh, and Frank C Park. Tangent bundle rrt: A randomized algorithm for constrained motion planning. *Robotica*, 34(1):202–225, 2016.

[7] R. Marchthaler and S. Dingler. *Kalman-Filter: Einführung in die Zustandsschätzung und ihre Anwendung für eingebettete Systeme*. Springer Fachmedien Wiesbaden, 2017.

[8] Commission implementing regulation (eu) no 923/2012 of 26 september 2012. *OJ*, L 281:14, 2013-10-13.

[9] Commission implementing regulation (eu) 2019/947 of 24 may 2019. *OJ*, L 152:45–71, 2019-06-11.

[10] Lester Eli Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. pages 497–516.